동국대학교

Lab 0 알고리즘 레포트

|  |  |
| --- | --- |
| 이름 | 한윤수 |
| 학과 | 컴퓨터AI학부 |
| 학번 | 2021113405 |
| 제출일자 | 2025.09.04 |

문제 1.

다음 주어진 코드는 파이썬을 이용하여 작성된 stack 자료구조이다. 이를 이용하여 아래의 입력을 주었을 때 주어진 출력이 나오도록 구현하고 구현된 코드가 어떤 작업을 하는 코드인지를 설명 하시오.

**입력 예제**

9 10 15 4 \* + 20 \* 7 + +

**출력 예제**

현재 시점 포인트 위치 : 1

현재 데이터 상태 :

['9']

현재 시점 포인트 위치 : 2

현재 데이터 상태 :

['9', '10']

현재 시점 포인트 위치 : 3

현재 데이터 상태 :

['9', '10', '15']

현재 시점 포인트 위치 : 4

현재 데이터 상태 :

['9', '10', '15', '4']

현재 시점 포인트 위치 : 3

현재 데이터 상태 :

['9', '10', 60]

현재 시점 포인트 위치 : 2

현재 데이터 상태 :

['9', 70]

현재 시점 포인트 위치 : 3

현재 데이터 상태 :

['9', 70, '20']

현재 시점 포인트 위치 : 2

현재 데이터 상태 :

['9', 1400]

현재 시점 포인트 위치 : 3

현재 데이터 상태 :

['9', 1400, '7']

현재 시점 포인트 위치 : 2

현재 데이터 상태 :

['9', 1407]

현재 시점 포인트 위치 : 1

현재 데이터 상태 :

[1416]

최종 데이터 :1416

소스코드 및 설명

|  |
| --- |
| class stack1:      def \_\_init\_\_(self, maxindex):          self.point = 0          self.maxindex = maxindex          self.stack = []      def \_\_del\_\_(self):          print("종료 시점 포인트 위치 : " + str(self.point))          print("종료 시점 데이터 상태 : ")          print(self.stack)      def push(self, v):          self.point += 1          self.stack.append(v)      def pop(self):          self.point -= 1          temp = self.stack[self.point]          self.stack.pop()          return temp      def empty(self):          if not self.stack:              return True          else:              return False      def cur\_print(self):          print("현재 시점 포인트 위치 : " + str(self.point))          print("현재 데이터 상태 : ")          print(self.stack)  def main():      input = "9  10  15  4  \*  +  20  \*  7  + +"      list = input.split()  # 입력을 분할하여 리스트에 저장      stack = stack1(len(list))  # 리스트의 크기를 전달하며 스택 생성      for curr in list:  # 리스트의 크기만큼 반복          value = None          if curr.isdigit():  # 현재 값이 정수인 경우              value = curr          else:  # 현재 값이 연산자인 경우              num1 = int(stack.pop())              num2 = int(stack.pop())              # 스택 상단에 있는 두 숫자를 꺼내서 연산              if curr == "+":                  value = num1 + num2              else:                  value = num1 \* num2          stack.push(value)  # 계산된 값을 다시 스택에 저장          stack.cur\_print()      fin = stack.pop()      print(f"최종 데이터 :{fin}")  if \_\_name\_\_ == "\_\_main\_\_":      main() |

결과 화면

|  |
| --- |
|  |

문제 2.

[code 제출]두 개의 자연수 N, M을 입력 받은 뒤 random 함수를 통해 N개의 자연수 집합 A와 M개의 자연수 집합 B를 만들어내는 프로그램을 작성하시오. 또한, 두 집합 A 와 B의 합집합과 교집합을 아래의 예제와 같이 출력해주는 프로그램을 작성하고 그 결과를 출력하시오. 마지막으로 합집합과 교집합을 찾아내는데 사용된 알고리즘의 원리를 설명하시오. (단, 파이썬의 set, intersection함수를 사용하면 안된다.)

**입력 예제**

11 12

**출력 예제**

집합 A – 1, 3, 4, 5, 9, 11, 19, 22, 26, 32, 55 집합 B – 3, 5, 7, 8, 9, 11, 12, 21, 32, 40, 42, 43 합집합 – 1, 3, 4, 5, 7, 8, 9, 11, 12, 19, 21, 22, 26, 32, 40, 42, 43, 55 교집합 – 3, 5, 9, 11, 32

소스코드 및 설명

|  |
| --- |
| import random  def main():      my\_input = "11 12"  # 입력 예제      sizes = my\_input.split()      size\_n = int(sizes[0])  # 집합 A의 크기 n      size\_m = int(sizes[1])  # 집합 B의 크기 m      lists = [[] for \_ in range(4)]  # 집합 A, 집합 B, 합집합, 교집합      # lists[0] = [1, 3, 4, 5, 9, 11, 19, 22, 26, 32, 55] # 집합 A 예제      # lists[1] = [3, 5, 7, 8, 9, 11, 12, 21, 32, 40, 42, 43] # 집합 B 예제      lists[0] = random.sample(range(1, 101), size\_n)  # 집합 A에 난수 배정      lists[1] = random.sample(range(1, 101), size\_m)  # 집합 B에 난수 배정      lists[2] = getUnion(lists[0], lists[1])  # 합집합      lists[3] = getIntersection(lists[0], lists[1])  # 교집합      titles = ["집합 A", "집합 B", "합집합", "교집합"]  # 각 집합에 대한 제목      lines = []  # 최종 출력문      for one\_title, one\_list in zip(titles, lists):  # 반복을 통해 출력문 생성          lines.append(f"{one\_title} - {one\_list}")      print("\n".join(lines))  # 정답 출력  # 합집합을 구하는 함수  def getUnion(setA, setB):      setU = setB.copy()  # 합집합, setB 복사      for val in setA:          if val not in setU:  # 값이 중복되지 않으면              setU.append(val)  # 합집합에 추가      return sorted(setU)  # 교집합을 구하는 함수  def getIntersection(setA, setB):      setI = []  # 교집합      for val in setA:          if val in setB:  # 값이 중복되면              setI.append(val)  # 교집합에 추가      return sorted(setI)  if \_\_name\_\_ == "\_\_main\_\_":      main() |

결과 화면

|  |
| --- |
|  |